yang_seonhyeHW31

Part 0

Setup

```
library(faraway)
data(pima)
```

Clean up the data

```
pima$glucose[pima$glucose == 0] <- NA
pima$diastolic[pima$diastolic == 0] <- NA
pima$triceps[pima$triceps == 0] <- NA
pima$insulin[pima$insulin == 0] <- NA
pima$bmi[pima$bmi == 0] <- NA
pima$diabetes[pima$diabetes == 0] <- NA
pima$age[pima$age == 0] <- NA

pima <- pima[complete.cases(pima),]
```

Here we remove the rows that have cases of zeroes.

```
print("Pregnant: ")
```

```
## [1] "Pregnant: "
```

```
quantile(pima$pregnant)
```

```
##   0%  25%  50%  75% 100%
##    0    1    2    5   17
```

```
print("Glucose: ")
```

```
## [1] "Glucose: "
```

```
quantile(pima$glucose)
```

```
##   0%  25%  50%  75% 100%
##   56   99  119  143  198
```

```
print("Diastolic: ")
```

```
## [1] "Diastolic: "
```

```
quantile(pima$diastolic)
```

```
##   0%  25%  50%  75% 100%
##   24   62   70   78  110
```

```
print("Triceps: ")
```

```
## [1] "Triceps: "
```

```r
quantile(pima$triceps)
```

```
##   0%  25%  50%  75% 100%
##    7   21   29   37   63
```

```r
print("Insulin: ")
```

```
## [1] "Insulin: "
```

```r
quantile(pima$insulin)
```

```
##     0%    25%    50%    75%   100%
##  14.00  76.75 125.50 190.00 846.00
```

```r
print("BMI: ")
```

```
## [1] "BMI: "
```

```r
quantile(pima$bmi)
```

```
##   0%  25%  50%  75% 100%
## 18.2 28.4 33.2 37.1 67.1
```

```r
print("Diabetes: ")
```

```
## [1] "Diabetes: "
```

```r
quantile(pima$diabetes)
```

```
##      0%     25%     50%     75%    100%
## 0.08500 0.26975 0.44950 0.68700 2.42000
```

```r
print("Age: ")
```

```
## [1] "Age: "
```

```r
quantile(pima$age)
```

```
##   0%  25%  50%  75% 100%
##   21   23   27   36   81
```

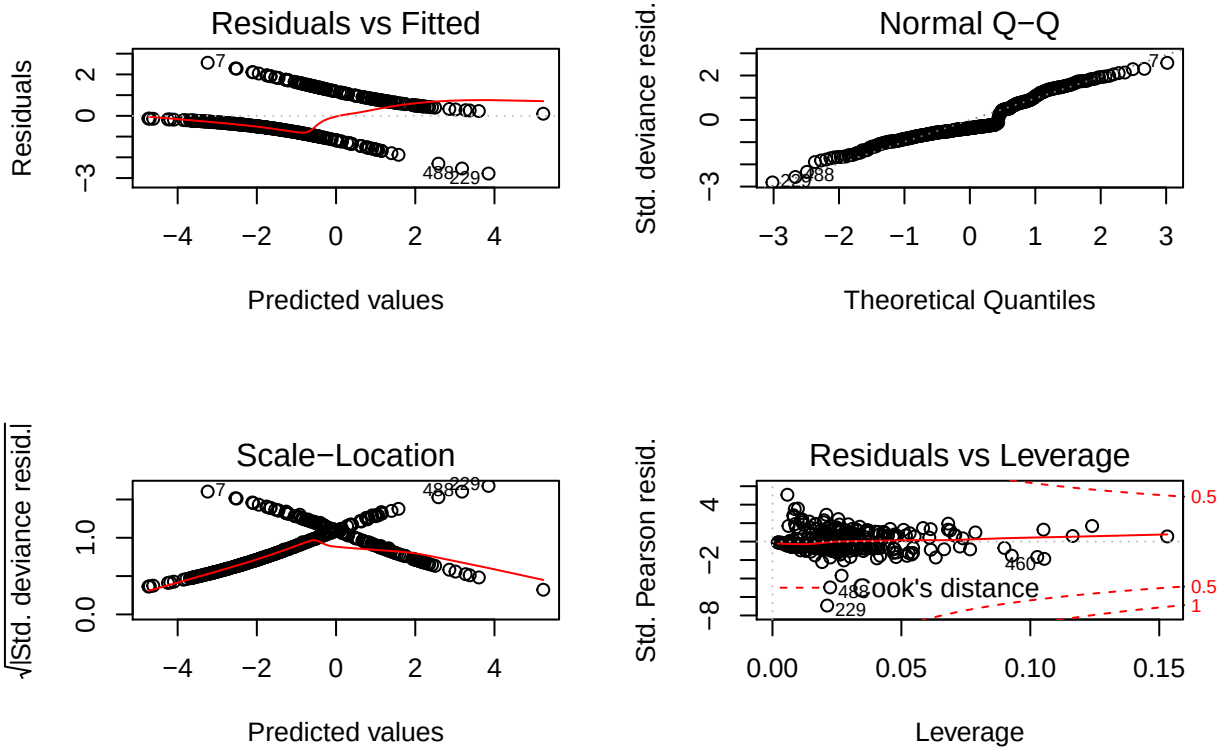
Part 1

```r
model <- glm(test ~ pregnant + glucose + diastolic + triceps + insulin + bmi + diabetes + age, data = pi
summary(model)
```

```
##
## Call:
## glm(formula = test ~ pregnant + glucose + diastolic + triceps +
##     insulin + bmi + diabetes + age, family = binomial(), data = pima)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7823  -0.6603  -0.3642   0.6409   2.5612
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.004e+01  1.218e+00  -8.246  < 2e-16 ***
```

```
## pregnant     8.216e-02  5.543e-02   1.482  0.13825
## glucose      3.827e-02  5.768e-03   6.635 3.24e-11 ***
## diastolic   -1.420e-03  1.183e-02  -0.120  0.90446
## triceps      1.122e-02  1.708e-02   0.657  0.51128
## insulin     -8.253e-04  1.306e-03  -0.632  0.52757
## bmi          7.054e-02  2.734e-02   2.580  0.00989 **
## diabetes     1.141e+00  4.274e-01   2.669  0.00760 **
## age          3.395e-02  1.838e-02   1.847  0.06474 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 498.10  on 391  degrees of freedom
## Residual deviance: 344.02  on 383  degrees of freedom
## AIC: 362.02
##
## Number of Fisher Scoring iterations: 5
```

```
par(mfrow=c(2,2))
plot(model)
```



From the results of this model, we can see that the coefficients that have a p value more significant than 0.05, we have the coefficients: glucose, bmi, and diabetes.

Part 2

```
stepModel <- step(model, direction = "backward")
```

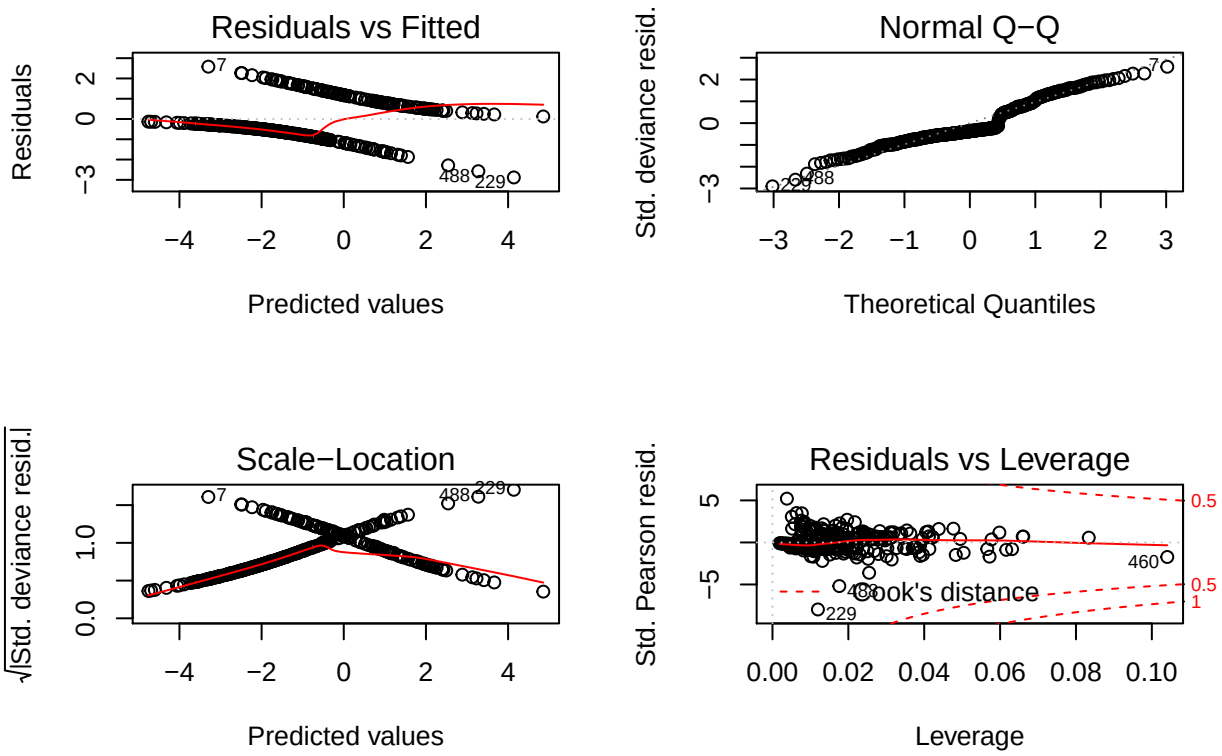
```
## Start:  AIC=362.02
## test ~ pregnant + glucose + diastolic + triceps + insulin + bmi +
##     diabetes + age
##
##             Df Deviance    AIC
## - diastolic  1   344.04 360.04
## - insulin    1   344.42 360.42
## - triceps    1   344.45 360.45
## <none>           344.02 362.02
## - pregnant   1   346.24 362.24
## - age        1   347.55 363.55
## - bmi        1   350.89 366.89
## - diabetes   1   351.58 367.58
## - glucose    1   396.95 412.95
##
## Step:  AIC=360.04
## test ~ pregnant + glucose + triceps + insulin + bmi + diabetes +
##     age
##
##            Df Deviance    AIC
## - insulin   1   344.42 358.42
## - triceps   1   344.46 358.46
## <none>          344.04 360.04
## - pregnant  1   346.24 360.24
## - age       1   347.60 361.60
## - bmi       1   351.28 365.28
## - diabetes  1   351.67 365.67
## - glucose   1   397.31 411.31
##
## Step:  AIC=358.42
## test ~ pregnant + glucose + triceps + bmi + diabetes + age
##
##            Df Deviance    AIC
## - triceps   1   344.89 356.89
## <none>          344.42 358.42
## - pregnant  1   346.74 358.74
## - age       1   347.87 359.87
## - bmi       1   351.32 363.32
## - diabetes  1   351.90 363.90
## - glucose   1   411.11 423.11
##
## Step:  AIC=356.89
## test ~ pregnant + glucose + bmi + diabetes + age
##
##            Df Deviance    AIC
## <none>          344.89 356.89
## - pregnant  1   347.23 357.23
## - age       1   348.72 358.72
## - diabetes  1   352.72 362.72
```

```
## - bmi       1   360.44 370.44
## - glucose   1   411.85 421.85
```

```
summary(stepModel)
```

```
##
## Call:
## glm(formula = test ~ pregnant + glucose + bmi + diabetes + age,
##     family = binomial(), data = pima)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8827  -0.6535  -0.3694   0.6521   2.5814
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -9.992080   1.086866  -9.193  < 2e-16 ***
## pregnant     0.083953   0.055031   1.526 0.127117
## glucose      0.036458   0.004978   7.324 2.41e-13 ***
## bmi          0.078139   0.020605   3.792 0.000149 ***
## diabetes     1.150913   0.424242   2.713 0.006670 **
## age          0.034360   0.017810   1.929 0.053692 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 498.10  on 391  degrees of freedom
## Residual deviance: 344.89  on 386  degrees of freedom
## AIC: 356.89
##
## Number of Fisher Scoring iterations: 5
```

```
par(mfrow=c(2,2))
plot(stepModel)
```



Looking at the residual plots for this AIC Step model when compared to the full model, we can see there is very little difference. However, it is interesting to note that the AIC Step model does include the age and pregnant variables, indicating that it thought they were significnat enough to contibute to the model. ## Part 3

```
exp(coef(stepModel))
```

```
##  (Intercept)     pregnant      glucose          bmi     diabetes
## 4.576094e-05 1.087578e+00 1.037130e+00 1.081273e+00 3.161077e+00
##          age
## 1.034957e+00
```

Seeing these odds-ratios, we can glean some information from the data. The odds-ratio tells us the factor the odds increase of the predicted variable by a one unit increase in the coefficient variable. We can see that diabetes has the highest odds-ratio, with a value of 3.161. This means for any 1 unit increase in diabetes, the chance of having diabetes (test) increases by a factor of 3.161. In addition to this, we can see that age had the smallest value of 1.034, so the chance of having diabetes changes by a factor of 1.034 with a one unit increase of diastolic.

Part 4

```
predictions <- predict(stepModel, pima) > .5
real <- pima$test > .5

correct <- 0
```

```
for (i in seq(1, length(predictions))) {
  if (predictions[[i]] == real[[i]]) {
    correct <- correct + 1
  }
}

(correct/length(predictions)) * 100
```

```
## [1] 78.31633
```

We can see here that our Step AIC model performed well, wtih an accuracy of 78.316%

Part 5

```
positivepositive <- 0
falsepositive <-0
negativenegative <- 0
falsenegative <- 0

for (i in seq(1, length(predictions))) {
  if (predictions[[i]] == TRUE && real[[i]] == TRUE) {
    positivepositive <- positivepositive + 1
  }
  else if (predictions[[i]] == TRUE && real[[i]] == FALSE) {
    falsepositive <- falsepositive + 1
  }
  else if (predictions[[i]] == FALSE && real[[i]] == FALSE) {
    negativenegative <- negativenegative + 1
  }
  else if (predictions[[i]] == FALSE && real[[i]] == TRUE) {
    falsenegative <- falsenegative + 1
  }
}

positivepositive
```

```
## [1] 65
```

```
falsepositive
```

```
## [1] 20
```

```
negativenegative
```

```
## [1] 242
```

```
falsenegative
```

```
## [1] 65
```

```
positivepositive/(positivepositive + falsepositive) * 100
```

```
## [1] 76.47059
```

```
negativenegative/(negativenegative + falsenegative) * 100
```

```
## [1] 78.82736
```

We can see from the results above, that when predicting a postive test result, our Step AIC model had a 76.47% truly positive rate, and when predicting a negative test result, it had a 78.82% truly negative result.